



Saku Viitalähde

Windows Phone -pelin kehitys ja julkaisu

Windows Phone -pelin kehitys ja julkaisu

Saku Viitalähde
Opinnäytetyö
Lukukausi Kevät vuosi 2013
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, Ohjelmistokehitys

Tekijä(t): Saku Viitalähde
Opinnäytetyön nimi: Windows Phone pelin kehitys ja julkaisu
Työn ohjaaja(t): Markku Rahikainen
Työn valmistumislukukausi ja -vuosi: Kevät 2013 Sivumäärä: 26

Opinnäytetyön tavoitteena oli kehittää ja edistää olemassa olevan Windows Phone -pelin alkua. Pelistä oli olemassa pieni versio, josta ohjelmaa lähdettiin kehittämään eteenpäin. Tarkoitukseni oli suunnitella ja toteuttaa peliin lisää sisältöä, korjata mahdollisia vikoja ja julkaista peli puhelimen kaupassa.

Projektin aikana oli tarkoitus tehdä useampi päivitys puhelinkauppaan sekä saada luotua selkeät ohjeet, miten julkaisu kauppaan tapahtuu ja mitä silloin pitää ottaa huomioon.

Projektin tarjoava yritys sai pelin kauppaan sekä päivitettyä sitä useamman kerran. Peliin saatiin tuotua projektin aikana uutta sisältöä ja monia vikoja saatiin korjattua. Projektin aikana kohdattiin muutamia ongelmia, jotka hidastivat hie-
man pelin kehitystä, mutta nekin selvitettiin. Projektin aikana tutustuttiin myös kaupassa julkaisuun ja sen asettamiin rajoihin. Kokemusten perusteella ohjeet julkaisulle saatiin laadittua.

Tulokseksi saatiin peli, jota on nyt ladattu useita tuhansia kertoja ympäri maail-
maa. Tulevaisuudelle saatiin myös hyvät suunnitelmat pitkälle eteenpäin, jolloin pelin kehittäminen jatkossa olisi helpompaa.

Asiasanat:

Kirjoita 3–7 kpl työtäsi kuvaavaa asiasanaa. Käytä hyväksesi esimerkiksi yleistä suomalaista asiasanastoa (YSO)[http:// www.yso.fi/onto/ysa/ysa_juuri](http://www.yso.fi/onto/ysa/ysa_juuri)

SISÄLLYS

TIIVISTELMÄ	3
SISÄLLYS	4
SANASTO	5
1 JOHDANTO	6
2 (WINDOWS PHONE) PELIN KEHITYS	7
2.1 Vaatimukset	7
2.2 Suunnittelu	7
2.3 Määrittely	8
2.4 Toteutus	9
2.5 Ylläpito	10
3 JULKAISU MARKETPLACE:SSÄ	12
3.1 Vaatimukset	12
3.2 Julkaisu	13
3.3 Seuranta ja päivittäminen	13
3.4 Maakohtaiset lisenssit	15
4 CASTLE DEFENDERS	16
4.1 Päävalikko	16
4.2 Pelaaminen	17
4.3 Pelin rakenne	18
4.4 Pelin erikoisloitsut	18
4.5 Pelissä ammunta	19
4.6 Pelin viholliset	20
4.7 Pelin Graafinen toteutus	21
4.8 Pelin kehitys	21
5 YHTEENVETO	22
LÄHTEET	23

SANASTO

Dev Center	Dev Center eli Developer Center on ohjelmistokehittäjille tehty sivusto oman ohjelman seurantaan ja hallintaan. (1.)
Microsoft	Microsoft Corporation on yhdysvaltalainen maailman suurin ohjelmistoalan yritys. (2.)
Scrum	Scrum on projektinhallinnan viitekehys, jota käytetään yleisesti ketterässä ohjelmistokehityksessä. (3.)
UML	UML-mallinnus on vuonna 1997 kehitetty graafinen mallinnuskieli, joka sisältää 13 erilaista kaaviota. Kaavioista kuudella kuvataan rakennetta, kolmella käyttäytymistä ja neljällä vuorovaikutusta. (4.)
Windows Marketplace	Windows Marketplace on Windowsin puhelimen virtuaalinen ohjelmistokauppa.
Windows Phone	Windows Phone on Microsoftin kehittämä matkaviestimiin tarkoitettu Windows-mobiilikäyttöjärjestelmä. (5.)

.

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli päästä kehittämään puhelinpeliä julkaisuun saakka sekä jatkaa sen päivittämistä ja ylläpitämistä julkaisun jälkeen. Idea oli saada kehittää ja päivittää jo olemassa olevaa peli-idea. Peli-idea tuli yritykseltä, jossa opinnäytetyöprojekti suoritettiin. Projektin tarkoituksena oli perehtyä siihen, miten peliä kehitetään, sekä mitä rajoitteita julkaisualusta antaa pelille.

Opinnäytetyön kirjoittaja oli osana pelinkehitysryhmää ja toimi ryhmän ainoana koodaajana. Ryhmään kuului myös graafikko ja päätösvaltainen yrityksen edustaja.

Opinnäytetyön toisena aiheena oli perehtyä pelin julkaisuun ja siihen asetettuihin rajoihin. Julkaisuvaiheessa projekti törmäsikin muutamaankin niistä.

Opinnäytetyön kirjallinen osuus käsittelee pelin kehitystä ja pelin julkaisua. Tarkoituksena on käydä kaikki vaiheet läpi, jotka pelin kehitykseen ja julkaisemiseen kuuluvat. Pelin kehitykseen on monenlaisia näkemyksiä, joista tässä esitellään yksi. Opinnäytetyössä pyritään käymään läpi kaikki julkaisuun liittyvät tapahtumat, ongelmat ja esteet.

2 WINDOWS PHONE -PELIN KEHITYS

2.1 Vaatimukset

Tässä vaiheessa ohjelmasta on olemassa vasta vaatimukset, jotka ohjelman pitäisi täyttää. Vaatimukset rajaavat ohjelman ja määrittävät sen käyttötarkoituksen. Vaatimukset voivat rajata myös ohjelmointikielen.

Vaatimusanalyysi on tärkeää tehdä tarkasti ohjelmiston tilaajan kanssa, jotta ohjelmasta varmasti saadaan juuri sellainen kuin on haluttu. Vaatimusanalyysin päätteeksi saadaan asiakasvaatimusdokumentti.

2.2 Suunnittelu

Pelisuunnittelu pitää sisällään monia osa-alueita, kuten ulkoasu, genre, pelin tyyli ja pelin maailma. Pelin suunnittelussa on tarkoitus saada kuva siitä, millaista peliä ollaan rakentamassa.(6.) Pelisuunnitelman aikana olisi hyvä olla useita ihmisiä jakamassa mielipiteitä ja ideoita, jotta näkemyksestä saataisiin riittävän laaja. Pelin suunnittelussa on hyvä käydä läpi kaikki asiat mahdollisimman tarkasti. Mitä tarkempi suunnitelma on, sitä helpompi muita vaiheita on tulevaisuudessa suorittaa.

Vajaa pelisuunnitelma voi aiheuttaa monimutkaisia ongelmia. Jos seuraaviin vaiheisiin siirrytään liian aikaisin, voi tulla vastaan asioita, joita kukaan ei ole ottanut omassa työssään huomioon. Tällaisesta voi seurata useiden kymmenien tuntien turhaa työtä. Näitä syistä täytyy muistaa, että kaikki pelikehityksen vaiheet ovat tärkeitä.

Pelisuunnittelussa pidetään yleensä suunnittelupalaveri. Palaverissa käydään kaikki suunniteltavat asiat läpi ja asiat kirjataan ylös jonkinlaisella dokumentointityylillä. Dokumentointityyli voi olla ihan mikä tahansa. Yleensä firmat käyttävät Word-dokumenttia, UML-kaavioita tai joskus jopa kynää ja paperia.

2.3 Määrittely

Ohjelmistomäärittely on asioiden tarkempaa tutkimista ja tarkkojen päätösten tekoa. Määrittelyvaiheessa kaikki päätökset kirjataan ylös, joiden pohjalta ohjelma lähdetään toteuttamaan. Määrittelyvaiheen lopussa määritelmän on oltava niin tarkka, että niillä ohjeilla ohjelma voidaan rakentaa. Määrittelyn tarkoituksena on saada mahdollisimman tarkka ja selkeä tekninen kuva ohjelmasta.

Määrittelyvaiheeseen kuuluvat myös laskelmat aikatauluista ja kustannuksista. Aikataulujen lisäksi määritellään myös ohjelman palaset sekä se, miten ja milloin niiden täytyy olla tehtynä. Tämä helpottaa kokonaisaikataulun seuraamista.

Määrittelyyn kuuluu myös päättää, kuka projektin aikana tekee mitäkin. Kaikille projektissa oleville jaetaan jo tässä vaiheessa omat työtehtävät. Näiden pohjalta työtä aletaan määrittelyn jälkeen toteuttaa.

Määrittelyvaihe on yritykselle hyvin tärkeä vaihe ohjelmistokehityksessä. Määrittelyvaihe on ohjelmistokehityksessä edullisin vaihe tehdä korjauksia yrityksen kannalta. Jos määrittelyvaiheessa jätetään jotakin määrittelemättä tai jokin asia määritellään väärin, virheistä voi koitua firmalle iso lasku toteutus- tai ylläpito-vaiheessa. Näistä syistä määriteltäessä on oltava erityisen tarkka.

Määrittelyn tarkoitus on myös päättää, millä kehitysperiaatteella ohjelmaa aletaan rakentaa. Suurimmat kehitysmallit ovat vesiputousmalli ja nykyään hyvin suosittu Scrum. Mallit eroavat täysin toisistaan. Vesiputousmallissa kaikki ohjelmistokehityksen vaiheet toteutetaan vain kerran, jonka jälkeen ohjelman pitäisi olla valmis(7.). Hyvällä suunnittelulla ja määrittelyllä vesiputousmalli säästää yritykselle huomattavasti aikaa ja rahaa. Scrum on uudempi kehitysmalli joka toimii useammalla kierrolla(8.). Scrumissa kaikki vaiheet käydään useasti läpi esimerkiksi viikon kierrolla. Nopea kierto tarjoaa mahdollisuuden muuttaa ohjelmiston suunnitelmaa ja määrittelyä myös kesken ohjelman teon. Scrum myös mahdollistaa varmemman ohjelmistontuotannon virheen tai muutoksen sattuessa.

Määrittelyssä dokumentointi on tärkeässä roolissa. Määrittelyyn on yritetty kehittää toimivia työkaluja jo vuosia. Nykyään UML-kaaviot ovat yleisiä(9.). UML on kehitetty järjestelmä, jolla voidaan luoda kaikki ohjelmistokehitykseen tarvittavat kaaviot ja kuvaajat. Määrittelyvaiheessa hyvään dokumentaatioon kuuluvat ainakin UML-kaavio, luokkakaavio, komponenttikaavio ja sekvenssikaavio. UML-kaavio näyttää ohjelman kaikki osat ja niiden liittymisen toisiinsa. Luokkakaavio on tarkoitus on määrittää koko ohjelmiston luokkarakenne ja niiden yhteistoimintaa. Komponenttikaavio näyttää miten ja mitkä ohjelmakomponentit ovat keskenään yhteydessä. Komponentit ovat luokkia isompia ohjelmapaloja. Sekvenssikaavio on taas vuorovaikutuskaavio, josta voidaan nähdä paljon pintaa syvemmälle. Sekvenssikaavio kuvaa ohjelmiston olioiden vuorovaikutusta ohjelmassa.

Näitä kaavioita hyväksikäyttäen ja muulla tapaa dokumentoiden saadaan hyvä määrittely ohjelmaa varten. Kaavioiden lisäksi yritykset käyttävät muiden asioiden kirjaamiseksi myös Excel- ja Word-dokumentteja. Näihin dokumentteihin kirjataan aikataulut, ohjelman vaatimukset sekä vastuualueet. Hyvän määrittelyn jälkeen on helppo siirtyä toteutukseen.

2.4 Toteutus

Windows Phonelle ohjelmoidaan Microsoft Visual Studiolla. Peliä toteuttaessa on valittavissa joko XNA-pohja tai Windowsin uudempi Silverlight-pohja. Molemmat ohjelmatyypit toimivat myös uusissa puhelinmalleissa, mutta vain Silverlight mahdollistaa muutamien visuaalisten ominaisuuksien käyttöä.

Ohjelmaa rakennettaessa on huolehdittava siitä, että ohjelmointi tehdään pala palalta määrittelyn mukaan. Usein ohjelmoinnissa voidaan törmätä ongelmaan, jota ei määrittelyssä ole huomattu. Tästä syystä nykyinen Scrum-kehitysmalli toimii paljon edeltäjänsä varmemmin, sillä ongelma voidaan korjata jo seuraavassa palaverissa.

Peliä ohjelmoitaessa pelistä tehdään yleensä ensin pieni runko. Tämä on kuten pelimoottori, jossa toimivat pelin perusmekaniikat, mutta johon voidaan tuoda

helposti lisää uutta sisältöä. Kun pelimoottori on valmis, sillä voidaan luoda demo. Tämä mahdollistaa mainonnan aloittamisen jo kauan ennen kuin ohjelma on oikeasti valmis. Moottorin rakentelun jälkeen sen päälle aletaan rakentaa tarinaa ja tarinan kulkua. Peliin luodaan kenttiä tai muuta sisältöä, joka siihen on alun perin suunniteltu. Viimeistään tässä vaiheessa pelinkehitykseen astuu mukaan graafinen suunnittelu. Graafikko mahdollistaa menu-valikon ja visuaalisen sisällön luomisen peliin. Kun pelissä on riittävästi toiminnallisuutta ja grafiikkaa, siihen aletaan tuoda ääniä. Äänimaailman luominen on äänimiehen tehtävä. Efektitäännet kuuluvat olennaisesti pelimaailmaan. Kun jotakin tapahtuu, siitä kuuluu tätä asiaa kuvaava ääni. Lisäksi äänimies suunnittelee pelintaustamuusikkejä. Suurissa pelitaloissa ääniä voi olla tekemässä kokonaisia orkestereita, mutta yleensä pienissä puhelinpeleissä niitä toteuttaa vain yksi henkilö.

Pelin toteutuksen ollessa siinä vaiheessa, että peliin on luotu määriteltä sisältö, peliä voidaan lähteä viemään eteenpäin. Ensimmäiseksi on kuitenkin tarkistettava, että peli on Windowsin ohjelmistokauppaan sopiva. Windows määrittelee sivuillaan siihen tarkat ohjeet, joihin palataan julkaisuasioissa. Kun peli on saatu näiltäkin osin kuntoon, se voidaan lähettää Windowsin laadunvalvontaan, joka tarkastaa ohjelman. Jos ongelmia ei tule, peli ilmestyy kauppaan noin viikon kuluttua lähetyksestä.

Pelin kehitys ei kuitenkaan pääty tähän. Usein peleistä löytyy julkaisun jälkeenkin vielä virheitä tai peliä halutaan muulla tapaa päivittää. Peli on siis siirtynyt ylläpitovaiheeseen.

2.5 Ylläpito

Ohjelmistojen ylläpitotyylejä on monenlaisia. Joskus ylläpitoon kuuluvat esimerkiksi serveripalvelut, tiedostonjako, päivittäminen ja käyttäjien hallinta. Tästä syystä ylläpitotehtävät voivat vaihdella suuresti. Seuraavat ylläpitotoimet olivat tarpeellisia projektissa.

Puhelinpeliympäristössä ylläpito keskittyy lähinnä pelin kokoaikaan kehitteemiseen ja päivittämiseen. Peliä on hyvä päivittää kuukausittain, koska tiheä päi-

vittäminen lisää market-näkyvyyttä. Peliin on tuotava jatkuvasti uutta sisältöä ja korjattava vikoja mikäli niitä ilmenee. Tasainen ja tiheä sisällön lisääminen pitää pelaajien mielenkiinnon yllä.

Ylläpidon jakelupalveluista ja latauspalvelimista puhelinpelimaailmassa vastaa yleensä puhelimen kehittäjä, eli tässä tapauksessa Microsoft. Kun tuote on saatettu markettiin, Microsoft pitää huolen siitä, että peli on milloin tahansa ladattavissa ja näkyvillä marketissa.

Pelien päivittäminen tapahtuu Microsoftin kautta. Kun pelin uusi päivitetty koodi on tehty, se lähetään uudelleen Microsoftin laaduntarkastukseen. Läpäistyään tarkastuksen Microsoft korvaa vanhan version uudella ja laittaa sen markettiin jaettavaksi. Jos päivitys ei läpäise tarkastusta, Microsoft ilmoittaa virheistä ja pitää vanhan version marketissa näkyvillä.

Jos yritys päättää lopettaa pelin kehityksen ja haluaa vetää sen ulos marketista, se on helppo poistaa. Yritys voi omia tunnuksiaan käyttäen kirjautua Microsoftin palveluun ja poistaa oman tuotteen marketista. Sen jälkeen peli ei ole enää marketissa nähtävillä eikä Microsoftin ylläpidettävissä.

3 JULKAISU MARKETPLACESSA

Kun ohjelma Windows Phonelle on tehty, se on julkaistava Windowsin Marketplacessa, jotta se saadaan näkyviin kaikille käyttäjille. Tässä luvussa kerrotaan, miten julkaisu markettiin tapahtuu ja mitä kaikkea se sisältää. Windows asettaa ohjelmille tietyt rajat, joiden puitteissa on eletävä. Onnistuneen julkaisun jälkeen on tärkeää myös päivittää ja kehittää tehtyä ohjelmaa. Luvussa perehdytään myös Windowsin asettamiin rajoihin sekä maakohtaisiin lisensseihin. Kaikki asiat on tärkeää ottaa huomioon, jotta voidaan toteuttaa mahdollisimman hyvä ja laaja julkaisu kerralla.

3.1 Vaatimukset

Windows asettaa ohjelmilleen tiukat vaatimukset. Suurin osa vaatimuksista on jo lainalaisia tai muuten hyvän maun mukaisia(10.). Ohjelmavaatimuksia on kaikille osa-alueille, kuten ohjelman sisällölle, toiminnallisuudelle ja rajapinnoille puhelimen kanssa. Kaikki ohjelmavaatimukset ovat nähtävillä Microsoftin Dev Center -sivulla, jonne julkaisu tehdään. Ohjelmavaatimukset testataan ja ajetaan puhelimella.

Mikäli julkaisutestauksessa jokin vaatimus ei täyty, Microsoft lähettää raportin julkaisijalle ja kehottaa tätä korjaamaan vian. Jos ohjelma halutaan markettiin, on viat korjattava ja lähetettävä ohjelma uudelleen tarkastettavaksi. Ohjelman vaatimustarkastus käydään läpi myös jokaisen päivityksen yhteydessä. Päivityksissä pätevät samat säännöt kuin julkaisussakin. Viat ilmoitetaan raporttina ja ne on korjattava, jotta päivitys saadaan kaikkien nähtäville.

Eroten muista kehitysympäristöistä Windows tuo joitakin omia määrittämiä ohjelmiin. Ohjelmien on tietysti oltava Windowsin omaa tiedostotyyppiä sekä täysin yhteensopivia Windows Phonen kanssa. Windows asettaa vaatimuksia myös ohjelmistojen sisällölle. Puhelimen taaksepäin olevasta nuolinäppäimestä on aina voitava palata takaisin edelliseen näkymään. Windows on vaativa myös äänien ja musiikin käytössä ohjelmissa. Jos puhelimella kuunnellaan musiikkia

ohjelman käynnistyttyä, on musiikin annettava jatkua eikä ohjelma ei saa säätää Windowsin soittimesta tulevaa musiikkia millään tavalla.

Muitakin pieniä poikkeuksia eri kehittäjien väliltä voi löytyä, mutta suurimmat ohjelmamäärittelykset koskevat patentteja tai rekisteröityjä merkkejä sekä muuta selvästi poikkeuksellista tarjontaa, kuten brutaalia väkivaltaa tai seksuaalista väkivaltaa. Olennaisimmat vaatimukset koskevat kuitenkin Windows Phonea ja sen tekniikkaa.

3.2 Julkaisu

Ohjelma julkaistaan Dev Centerin kautta Microsoft Marketplacessa. Ohjelman julkaisuprosessi tehdään sivulla olevien ohjeiden mukaan. Ohjelmalle annetaan nimi ja kategoria. Ohjelmasta voi lisätä kuvia ja kirjoittaa kuvaavaa tekstiä. Kaikki annettu tieto tulee kaikkien näkyville markettiin. Lähetysvaiheessa ohjelmalle saa määritellä myös hakusanoja, jotka helpottavat ohjelman löytymistä marketista. Jo aluksi julkaisijan on päätettävä, paljonko ohjelman lataus maksaa vai jaetaanko sitä ilmaiseksi. Ohjelmaan liitetään myös hankitut lisenssit laajempaa levitystä varten. Joillakin mailla ja alueilla on omat lisenssinsä, jotka kannattaa hankkia.

Ohjelman julkaisussa testattavaksi lähetään ohjelman XAP-tiedosto. XAP-tiedosto on pakattu ohjelmatiedosto ja sisältää siis koko ohjelman. Julkaisussa Microsoft testaa ohjelman kokonaisvaltaisesti. Jos kaikki vaatimukset täyttyvät, ohjelma pääsee markettiin. Ohjelman testaus kestää noin viikon, jonka jälkeen saadaan tulos.

Ohjelman ollessa marketissa julkaisija saa päättää, onko ohjelma vielä kaikille nähtävillä vai piilossa. Kun julkaisija päättää asettaa ohjelman julkiseksi, se on ladattavissa puhelimella sekä nähtävillä myös marketin verkkosivuilla.

3.3 Seuranta ja päivittäminen

Ohjelman seuranta ja päivittäminen on helppoa Dev Centerin kautta. Sivusto tarjoaa helpon tavan seurata ohjelman latausmääriä. Latausmääriä voi katsella

eri aikaväleiltä sekä seurata ohjelman kasvua koko ajalta. Seurantatyökaluilla nähdään, mihin ohjelmaa on ladattu. Näistä voidaan nähdä minkälaisia pelejä kuluttajat lataavat missäkin maissa. Seurannassa kannattaa tarkkailla myös ohjelman kasvunopeutta. Kasvunopeus kertoo ohjelman suosion suunnan. Tasaisesti nouseva kasvunopeus on hyvä merkki. Jos kasvunopeuspiikki on jo mennyt, on syytä aloittaa päivitykset tai sisällön kehittäminen.

Ohjelmiin voi upottaa erilaisia seurantatyökaluja, joilla voidaan kerätä käyttäjien tietoja. Ohjelman sisäisillä välineillä voidaan seurata, kauanko ohjelmaa on käytetty tai kuinka käyttäjien ikä ja sukupuoli jakautuvat. Kerätyillä tiedoilla ohjelmaa voidaan kehittää halutulle kohderyhmälle sopivaksi.

Ohjelman päivittäminen on helppo tapa saada näkyvyyttä marketissa sekä pitää käyttäjät kiinni ohjelmassa. Marketplace antaa ilmaista näkyvyyttä niille ohjelmille, joita päivitetään usein. Siksi jatkuva ohjelman päivittäminen pieninä paloina kerrallaan on pienelle ohjelmistojenkehittäjälle tärkeä tapa saada näkyvyyttä.

Päivittäminen tehdään Dev Centerissä, johon ohjelman kehittäjä lähettää uuden päivitetyn ohjelman. Dev Center testaa taas ohjelman ja jos kaikki vaatimukset täyttyvät, ohjelman päivitys menee markettiin edellisen julkaisun tilalle. Päivityksen onnistumiseksi ohjelmaan on tultava jokin muutos ja sen versionumeron on kasvettava. Muutoksen ei tarvitse aina olla suuri, mikä mahdollistaa tasaisen ja tiheän päivityksen. Näkyvyyden säilyttämiseksi ohjelmaa olisi hyvä päivittää noin kerran kuudessa viikossa. Näkyvyys säilyy sitä paremmin, mitä useammin kehittäjä päivittää ohjelmaa. Päivityksestä tulee ilmoitus ohjelman käyttäjille, jotka näkevät uutta sisältöä päivittämällä ohjelman puhelimeensa.

3.4 Maakohtaiset lisenssit

Ohjelmaa markettiin ladattaessa sille voidaan asettaa lisenssejä, joilla näkyvyyttä voi lisätä. Ilman oikeita lisenssejä joidenkin maiden marketit eivät hyväksy ohjelmaa. Lisenssejä voidaan lisätä ohjelmaan myös päivitysten mukana. Markettiin voi saada seuraavia lisenssejä.

CERO	Japanilainen organisaatio, joka vastaa videopelien ja muiden ohjelmistojen ikäluokituksesta.(11.)
COB-AU	Australian ikäluokituslisenssi.(12.)
CSRR	Kiinan ja Taiwanin ikäluokituslisenssi.(13.)
DJCTQ	Brasilian ikäluokituslisenssi.
ESRB	Itsesäätelevä organisaatio, jonka ikäluokitusjärjestelmä videopeleille ja muulle multimedialle toimii Yhdysvalloissa ja Kanadassa.(14.)
FPB	Etelä-Afrikan ikäluokituslisenssi.(15.)
GRB	Game Rating Board, korealainen ikäluokitusjärjestelmä.(16.)
OFLC_NZ	Uuden-Seelannin ikäluokituslisenssi.(17.)
PEGI	PEGI on yleiseurooppalainen ikäluokitusjärjestelmä videopeleille ja muille ohjelmistoille.(18.)
PEGI_PT	Portugalin ikäluokituslisenssi.
USK	Saksalainen järjestö, joka myöntää ohjelmistoille ikäluokituksia.(19.)

4 CASTLE DEFENDERS

Castle Defenders on Windows Phone -peli. Pelin tavoitteena on suojella ruudun yläreunassa olevaa linnaa sitä kohti hyökkääviltä vihollisilta. Peliä kehitettiin pienessä ryhmässä graafikon ja äänimiehen kanssa. Yrityksellä oli valmiina pelin idea, jota projektissa lähdettiin kehittämään.

4.1 Päävalikko

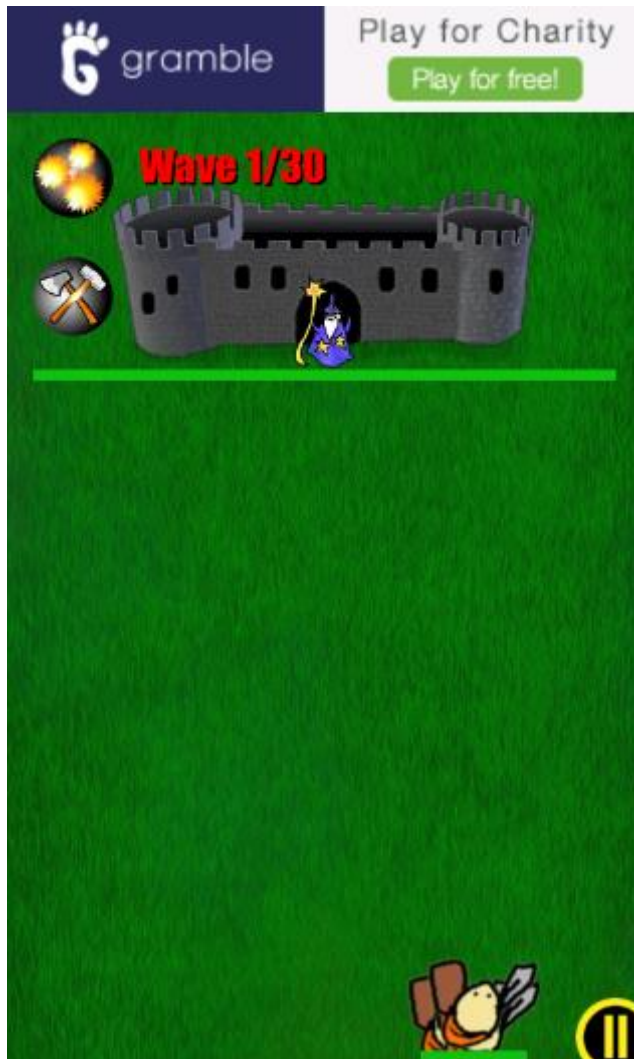
Pelin avauduttua tulee nähtäväksi päävalikko (kuva 1). Valikossa näkyy pelin logo, versio sekä useampi painike. Start-painikkeesta käyttäjä voi aloittaa pelaamisen. Päävalikossa on myös Help-painike, jota painamalla pelaaja näkee selkeät ohjeet pelin pelaamiseen. Päävalikko sisältää myös Exit-painikkeen, jolla pelistä voi poistua, sekä musiikin voimakkuuden säätöpainikkeet.



KUVA 1. Pelin päävalikosta.

4.2 Pelaaminen

Aloitettuaan pelin pelaaja saa nähtäväksi pienen tarinan, minkä jälkeen itse peli alkaa. Alussa pelaajan linna on näytön yläreunassa, ja vihollisia alkaa heti tulla ruudun alareunasta linnaa kohti (kuva 2). Pelaajan tehtävä on ampua linnasta vihollisia kohti ja tuhota ne. Pelin edetessä vihollismäärät kasvavat, ja vihollisista tulee vaikeampia tuhota.



KUVA 2. Pelin lähtötilanteesta.

4.3 Pelin rakenne

Pelissä edetään kymmenen hyökkäyksen paketeissa. Viimeisessä hyökkäyksessä eteen tulee suurempi ja voimakkaampi vihollinen. Peli muuttuu hieman aina, kun pelaaja on puolustanut linnaa kymmenen hyökkäyksen ajan. Hyökkäysten jälkeen tulee nähtäville uusi tarinan jatko, josta jatketaan seuraaviin hyökkäyksiin. Pelissä on tällä hetkellä kolme kymmenen hyökkäyksen pakettia. Ensimmäisen paketin jälkeen pelaaja saa käyttöönsä jääammuksen. Jääammus hidastaa vihollisia merkittävästi ja tekee niihin pientä vahinkoa. Toisen paketin jälkeen pelissä tulee talvi, jolloin pelaajan on selviydyttävä seuraavista hyökkäyksistä ilman hidastusta, koska talvella viholliset eivät hidastu jääammuksesta.

4.4 Pelin erikoisloitsut

Pelaajalla on käytössä myös kaksi apuvälinettä (kuva 3). Apuvälineitä voi käyttää vain kerran jokaisella pelikerralla. Sarjatuli on apuväline, joka toimii nopeasti suurienkin vihollismassojen tuhoamiseen. Sarjatuli kestää vain kuudenkymmenen panoksen ajan, jonka jälkeen ampuminen muuttuu takaisin normaaliksi. Toisena apuvälineenä pelissä toimii linnankorjaus. Jos linna on pelin aikana ottanut vahinkoa, sitä voidaan korjata. Vaikka korjausapuväline korjaa linnaa vain vähän, se on monissa tilanteissa elintärkeä pelin jatkumisen kannalta.



KUVA 3. Castle Defenders -pelin erikoisloitsut.

4.5 Ammunta pelissä

Pelissä ampuminen tapahtuu ruutua painamalla, jolloin linnasta lentää ammus kohti painettua pistettä. Ammus lentää painetun pisteen kautta suoraa linjaa ulos ruudusta. Jos ammus osuu viholliseen matkalla, se tekee siihen vahinkoa ja läpäisee vihollisen (kuva 4). Läpäiseminen mahdollistaa ampumisen takti-
koinnin, jolloin pelaajan tehtäväksi jää mahdollisimman hyvien ampumislinjojen löytäminen. Pelin viholliset tulevat ruudulle satunnaisia linjoja pitkin, joten am-
pumislinjoja ei voi opetella, vaan ne on mietittävä uudelleen tapauskohtaisesti.



KUVA 4. Pelitilanne pelistä.

4.7 Pelin graafinen toteutus

Peli on tarkoituksella tehty täysin kaksiulotteiseksi. Tyyliksi valittiin yksinkertainen ja helposti tuotettava grafiikka. Pelin koko graafinen ilme perustuu siis yksinkertaiseen kaksiulotteiseen grafiikkaan, jossa pelimaailma on luotu laittamalla erilaisia kuvia päällekkäin.

4.8 Pelin kehitys

Projektia aloitettaessa peliä ei oltu suuremmin suunniteltu tai määritelty. Projektin kehitys tapahtui lyhyissä pätkissä, joiden jälkeen pidettiin palaveri. Palaverissa suunniteltiin, mitä seuraavaksi tehdään, ja seuraavalla viikolla se toteutettiin. Mikäli ongelmia ilmeni, nostettiin asia pöydälle seuraavassa palaverissa ja mietittiin, miten asia ratkaistaan. Suurempia suunnitelmia tässä projektissa ei käytetty, vaan kaikki tehtiin aina viikon eteenpäin. Joitakin ongelmiakin tietysti ilmeni, mutta kaikki ongelmat saatiin aika nopeasti korjattua. Ohjelmistokehitystyylinä tämä toimintatapa muistuttaa etäisesti Scrum-mallia.

Projektissa käytetyn kehitysmallin etuna oli tietysti se, että peliä päästiin heti tekemään ja asia saatiin liikkeelle. Mallin ainoa huono puoli oli puute suunnittelussa. Projektin myöhemmissä vaiheissa kävi ilmi, että joitakin suunnittelun vaiheita on jäänyt huomiotta. Ongelmat olivat kuitenkin hyvin pieniä, joten projektia saatiin hyvin vietyä eteenpäin.

Projektin aikadokumentointia ei ollut. Palaverin jälkeen sovitut asiat kirjattiin sähköpostiin ja se lähetettiin asianomaisille. Näin kaikki pysyivät ajan tasalla siitä, mitä heidän kuuluu tehdä milläkin viikolla. Näin pienessä projektissa tällainen kehitysmalli toimii, mutta jos kyseessä olisi suurempi ja monimutkaisempi projekti, kannattaisi tehdä ensin hyvät suunnitelmat ja dokumentaatiot ennen projektin varsinaista aloitusta.

Projektin ohjelmointiin käytettiin Visual Studio 2010 -ohjelmaa ja Windows Phone SDK:ta. Ohjelmointikielenä projektissa oli C#, jolla koko ohjelman toiminta suoritettiin.

5 YHTEENVETO

Projektin ideana oli jatkaa olemassa olevaa peliprojektin alkua. Peliin oli tuotettava lisää sisältöä ja hankittava selkeät ohjeet tulevaisuutta varten, miten ohjelman julkaisu puhelinkauppaan tapahtuu ja mitä siellä vaaditaan.

Projektin alussa peli julkaistiin kaupassa pienten määrittelykorjausten jälkeen. Tästä eteenpäin peliä suunniteltiin ja toteutettiin sopivin ajanjaksoin. Pelissä havaittuja vikoja korjattiin sekä uutta sisältöä suunniteltiin ja toteutettiin.

Projektin aikana tuli jo olemassa olevien vikojen lisäksi myös uusia korjattavia. Viat saatiin korjattua, mutta ne hidastivat alkuperäistä kehitystahtia. Ohjelmaa päivitettiin usein, mikä takasi ilmaista kaupanäkyvyyttä.

Projektin aikana peliä ladattiin kaupasta kymmeniä tuhansia kertoja, minkä takia olisi ollut hyvä saada uutta sisältöä pelille mahdollisimman nopeasti. Sisällöllisessä päivitysnopeudessa oli kuitenkin annettava periksi ilmenneiden vikojen ansiosta.

Projektissa opin käytännössä, mitä ohjelmistotuotanto on, ja millaisia asioita siinä voi kohdata. Projektissa sain myös hyvää kokemusta XNA-sovelluksista ja siitä, miten niitä kehitetään. Oppia tuli myös ohjelmiston ja sen sisällön suunnittelussa sekä yleisessä ohjelmiston ongelman ratkaisussa. Projektin aikana opin monipuolisesti uutta pienemmästä ohjelmaprojektista.

LÄHTEET

1. Dev Center. 2013. Saatavilla: <http://developer.windowsphone.com/en-us>.
Hakupäivä 2.4.2013
2. Microsoft. 2013. Saatavilla: <http://fi.wikipedia.org/wiki/Microsoft>.
Hakupäivä 2.4.2013
3. Scrum. 2013. Saatavilla: <http://www.scrum.org/>. Hakupäivä 2.4.2013
4. UML-mallennus. 2013. Saatavilla: <http://fi.wikipedia.org/wiki/UML-mallinnus>. Hakupäivä 2.4.2013
5. Windows Phone. 2013. Saatavilla:
http://fi.wikipedia.org/wiki/Windows_Phone. Hakupäivä 2.4.2013
6. Pelisuunnittelu. 2013. Saatavilla: <http://fi.wikipedia.org/wiki/Pelisuunnittu>.
Hakupäivä 25.2.2013
7. Vesiputousmalli. 2013. Saatavilla:
<http://fi.wikipedia.org/wiki/Vesiputousmalli>. Hakupäivä 7.3.2013
8. Scrum. 2013. Saatavilla: <http://fi.wikipedia.org/wiki/Scrum>. Hakupäivä
7.3.2013
9. UML. 2013. Saatavilla: <http://users.metropolia.fi/~hakka/Ohma/Ohma-02.pdf>. Hakupäivä 7.3.2013
10. App certification requirements for Windows Phone. 2013. Saatavilla:
[http://msdn.microsoft.com/en-US/library/windowsphone/develop/hh184843\(v=vs.105\).aspx](http://msdn.microsoft.com/en-US/library/windowsphone/develop/hh184843(v=vs.105).aspx). Hakupäivä
23.3.2013
11. CERO. 2013. Saatavilla:
http://fi.wikipedia.org/wiki/Computer_Entertainment_Rating_Organizatio.
Hakupäivä 18.3.2013
12. Games for Windows Technical Requirements. 2013. Saatavilla:
[http://msdn.microsoft.com/en-us/library/windows/desktop/ee417691\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ee417691(v=vs.85).aspx). Hakupäivä
18.3.2013

13. CSRR. 2013. Saatavilla:
http://en.wikipedia.org/wiki/Computer_Software_Rating_Regulation. Hakupäivä 18.3.2013
14. ESRB. 2013. Saatavilla:
http://fi.wikipedia.org/wiki/Entertainment_Software_Rating_Board. Hakupäivä 18.3.2013
15. FPB. 2013. Saatavilla: <http://www.balancingact-africa.com/news/en/issue-no-579/web-and-mobile-data/app-and-music-store/en>. Hakupäivä 18.3.2013
16. GRB. 2013. Saatavilla <http://fi.wikipedia.org/wiki/GRB>. Hakupäivä 18.3.2013
17. OFLC_NZ. 2013. Saatavilla: <http://www.censorship.govt.nz/>. Hakupäivä 18.3.2013
18. PEGI. 2013. Saatavilla:
http://fi.wikipedia.org/wiki/Pan_European_Game_Information. Hakupäivä 18.3.2013
19. USK. 2013. Saatavilla:
http://fi.wikipedia.org/wiki/Unterhaltungssoftware_Selbstkontrolle. Hakupäivä 18.2.2013